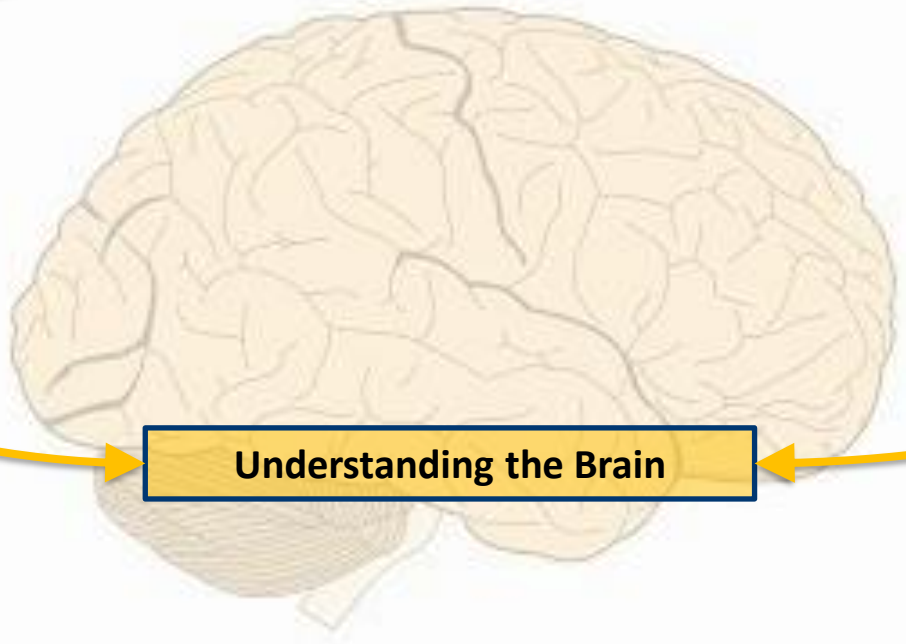
Exploring Novel Neuromorphic Computing Architectures with a Multi-Node FPGA System

Pierpaolo Perticaroli
INFN Roma 1 – APE Lab

BRAINSTAIN - CSN5 PROJECT
BRAIN Studies and Technologies for Artificial Intelligence and Neuroscience



WP1
Analysis pipelines
data processing, **data analysis**

WP2
Brain models
&
Simulations

Task 1
Algorithms & Methods

Task 2
Data Analysis

Task 3
Clinical Applications

Understanding the Brain

Hardware/Software co-design
of Computing Architectures

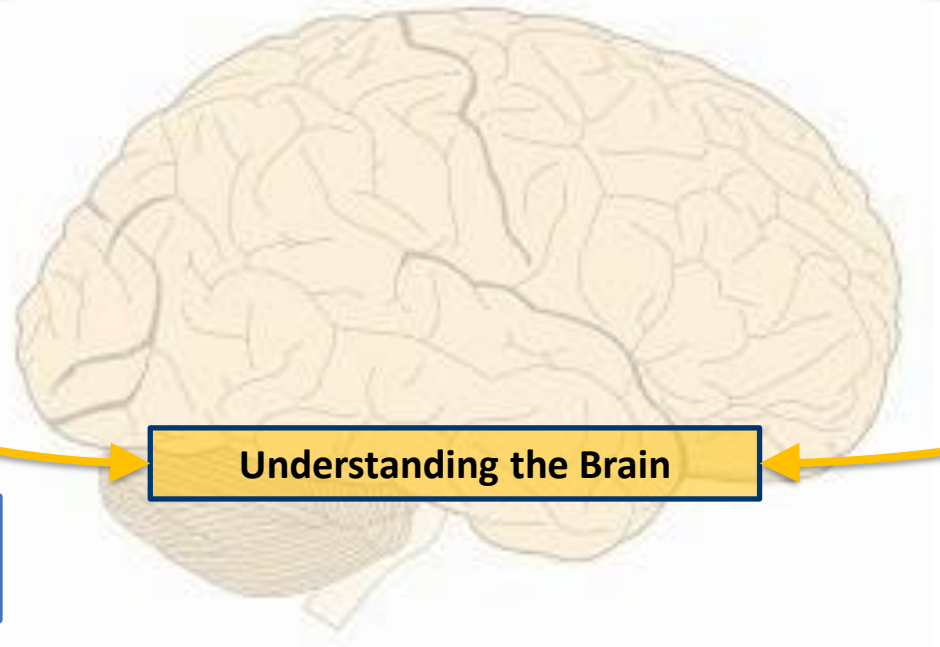
WP3
Design of Brain-inspired
Computing Architectures for AI

Task 1
From brain models to architecture

Task 2
Architectures prototyping

Task 3
Validation and Perf. Estimation

BRAINSTAIN - CSN5 PROJECT
BRAIN Studies and Technologies for Artificial Intelligence and Neuroscience



WP1
Analysis pipelines
data processing, **data analysis**

WP2
Brain models
&
Simulations

Task 1
Algorithms & Methods

Task 2
Data Analysis

Task 3
Clinical Applications

Understanding the Brain

Hardware/Software co-design
of Computing Architectures

WP3
Design of Brain-inspired
Computing Architectures for AI

Task 1
From brain models to architecture

Task 2
Architectures prototyping

Task 3
Validation and Perf. Estimation

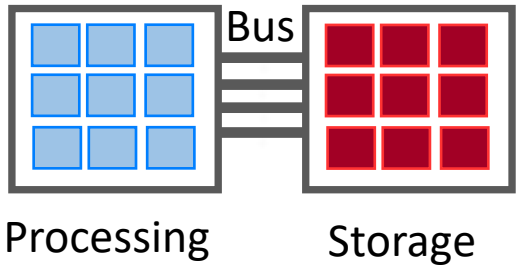
Neuromorphic Computing - Why mimic the Brain?

- **In-Memory Computing**. Co-location of processing and storage

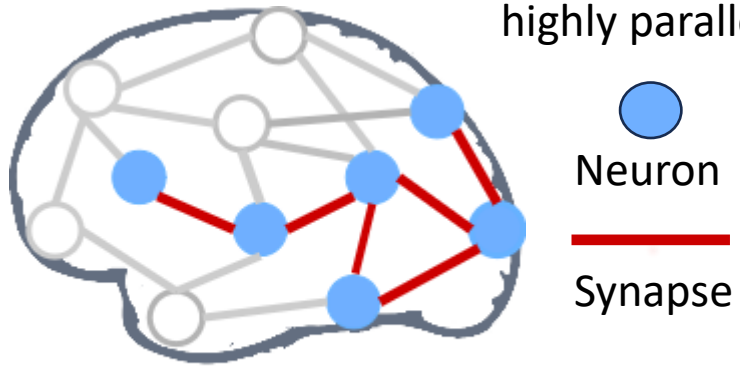
Traditional Von Neuman computing. The cost of data movement

Operation	Energy consumption
Addition of data	1x
Access data (onchip cache)	60x
Access data (offchip RAM)	3500x

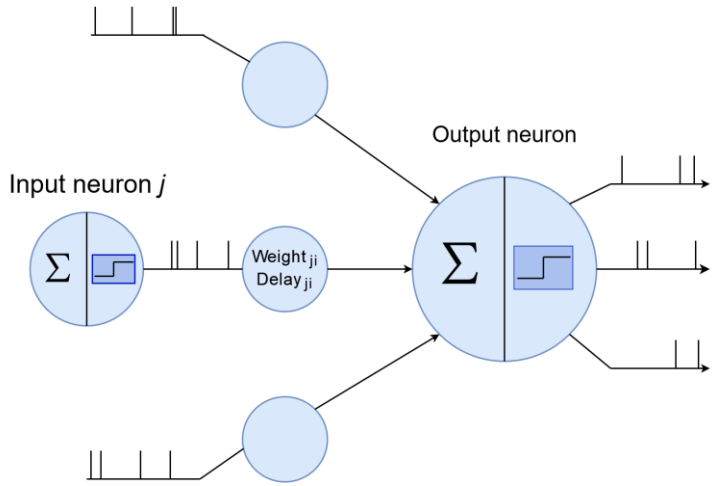
Sze et al, IEEE Custom Integrated Circuits Conference (2017)



Brain. Closely tied processing-storage, highly parallel



- Sparse, distributed information encoding through spikes
- Asynchronous, event-based computation relying on spikes time
- Local learning rules
  - No costly back-propagation



High energy efficiency, real-time processing, incremental learning

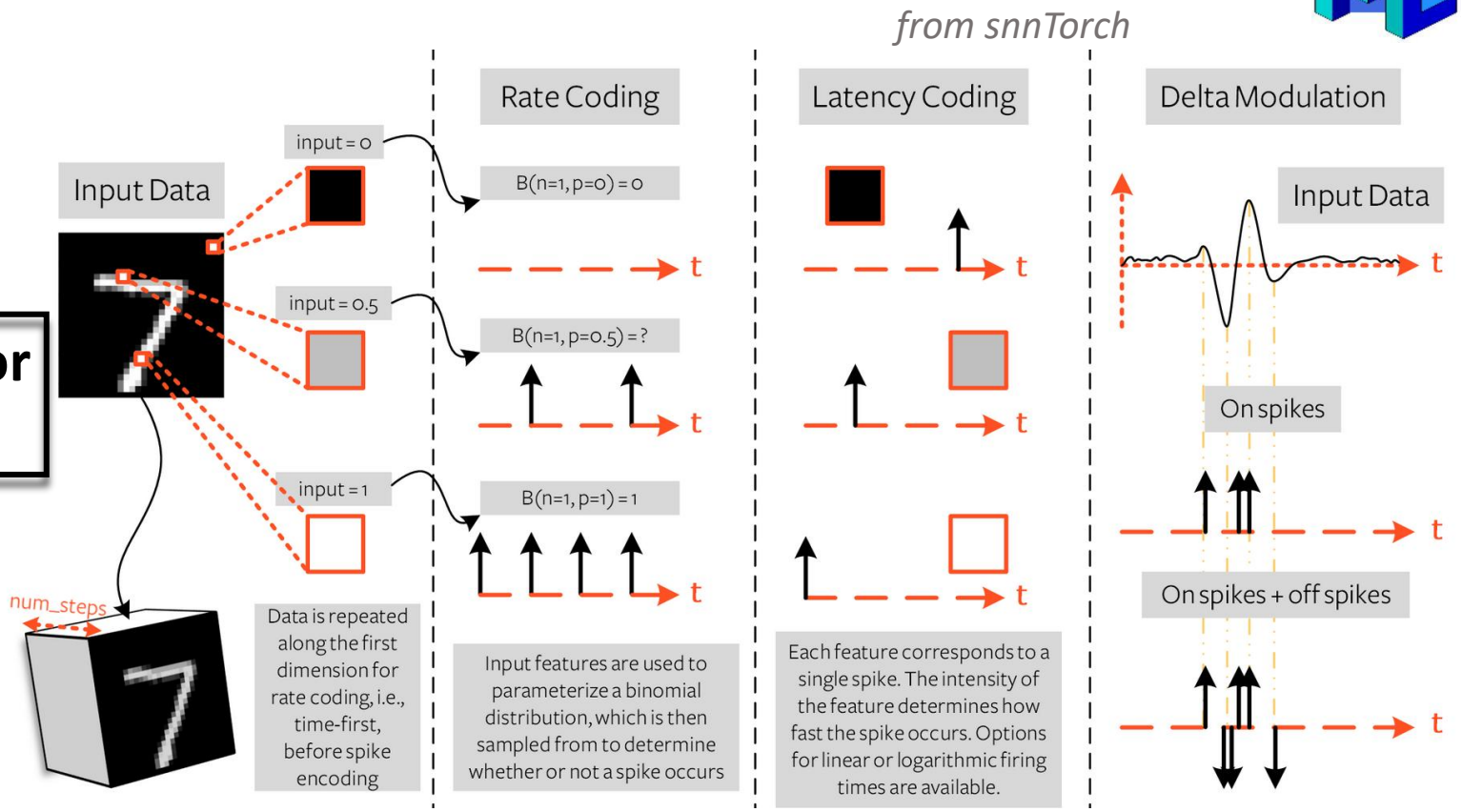
Spiking Neural Networks Applications

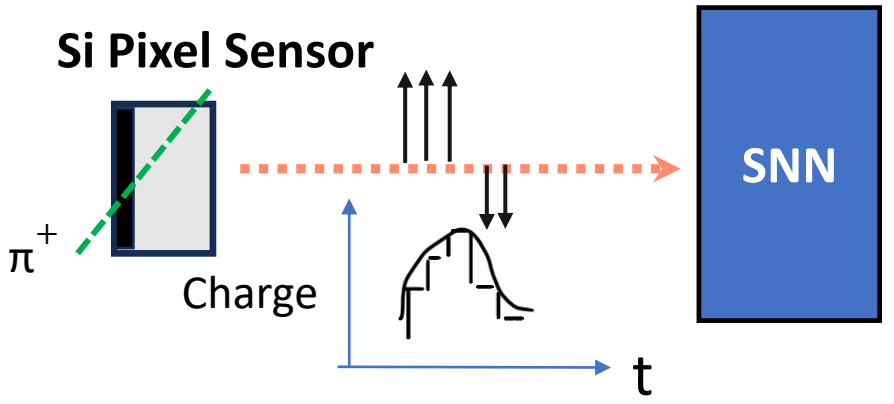
Energy efficiency,
Real-time learning

Ideal for edge Computing, on-sensor processing, dataflow applications

- **Any input can be encoded as spikes**
- Particularly effective when input is already sparse, event-based or is acquired as a time sequence

- Identify interesting applications in physics
  (particle physics sensors, anomaly detection, event cameras, time series data ...)





What we are working on

Multi-node FPGA system

- Modular, scalable and reconfigurable neuromorphic architecture

Our target

- **Accelerator platform for Spiking NN**

  Edge and dataflow applications in physics experiments

What we are working on

Multi-node FPGA system

- Modular, scalable and reconfigurable neuromorphic architecture
- Support for biologically realistic neuron dynamics
- Support for multi-compartment neuron models
- Deterministic, reproducible results

Our target

- **Accelerator platform for Spiking NN**

  Edge and dataflow applications in physics experiments

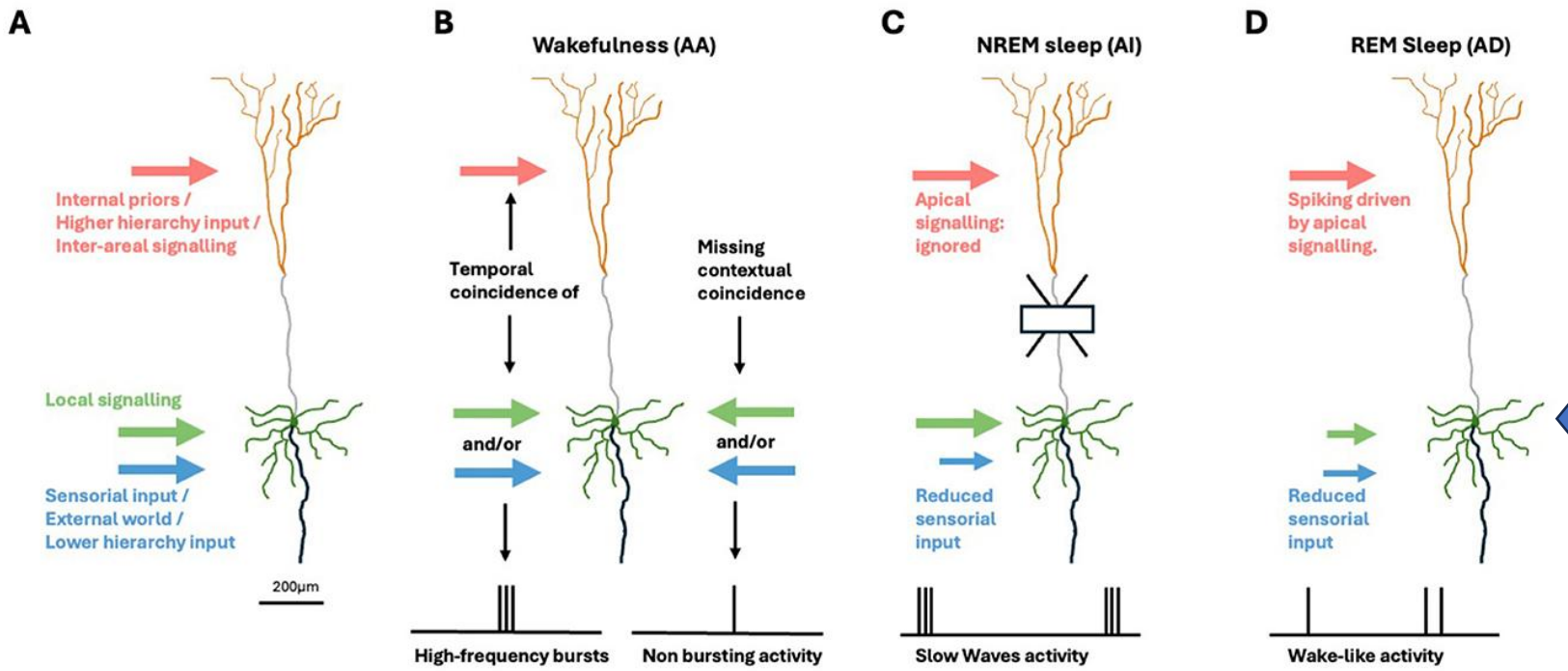
- **Neuroscience simulator for specific use cases**

  Explore application of novel bio-inspired AI

What we are working on

Multi-node FPGA system

- Modular, scalable and reconfigurable neuromorphic architecture
- Support for biologically realistic neuron dynamics
- Support for multi-compartment neuron models



Our target

- **Accelerator platform for Spiking NN**

  Edge and dataflow applications in physics experiments

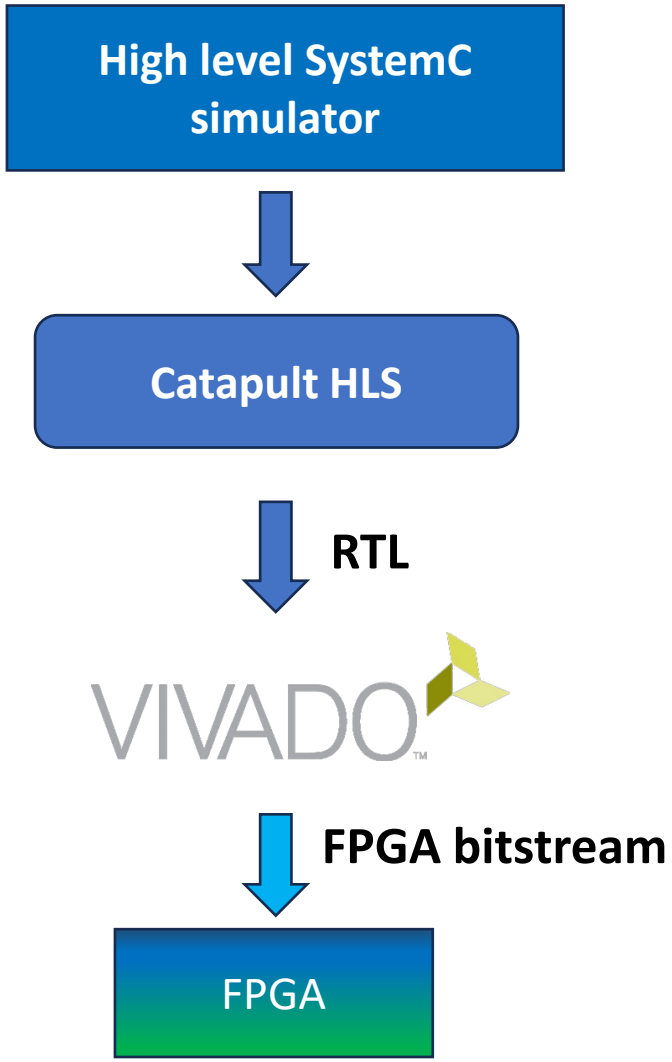
- **Neuroscience simulator for specific use cases**

  Explore application of novel bio-inspired AI

Focus on

Simplified two-compartment neuron with calcium dynamics capturing brain-state specific apical-amplification, -isolation and –drive Pastorelli et Al. 2025

Architecture Design – Workflow and Tools



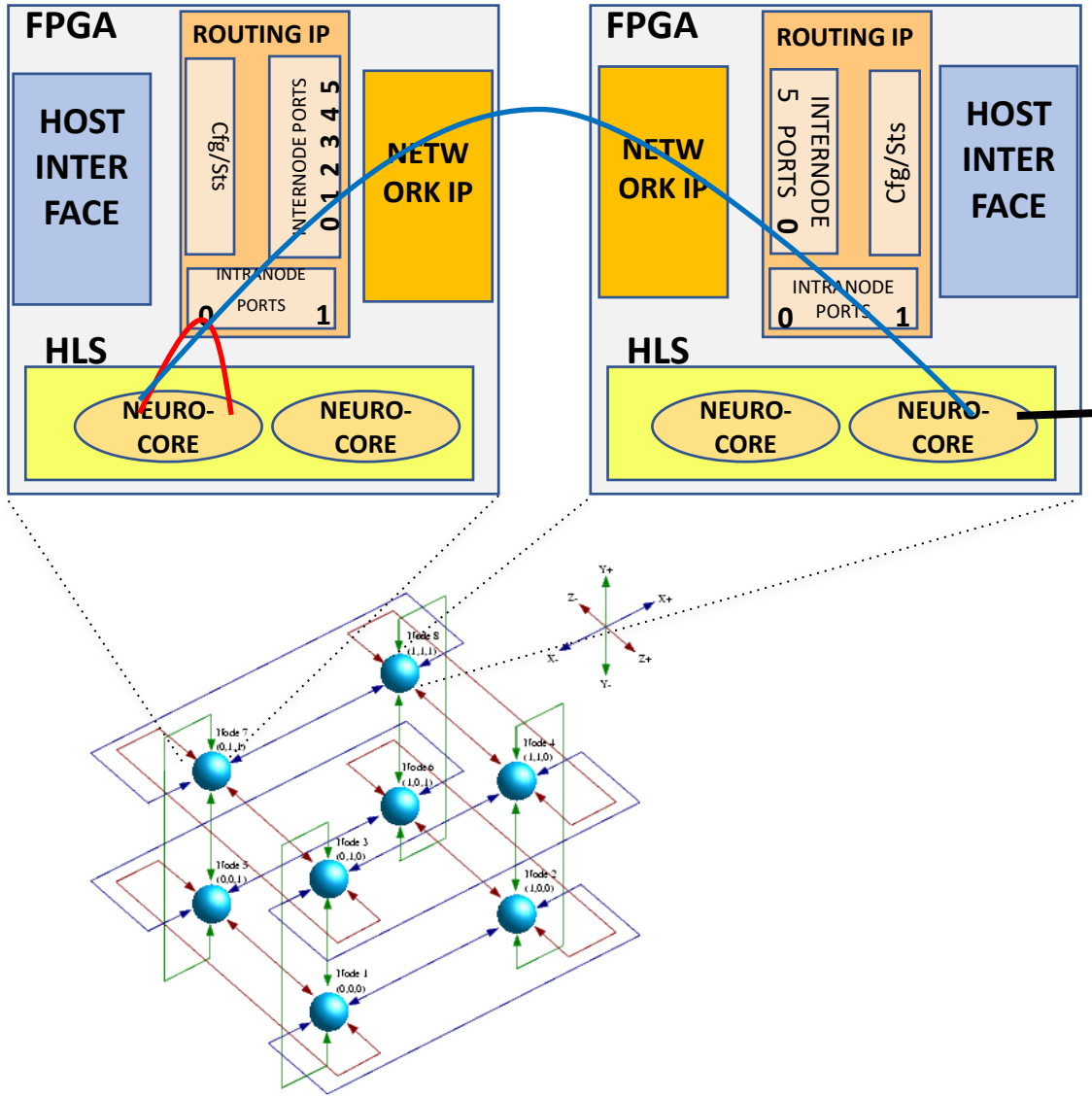
- **SystemC (C++ based event-driven) architecture simulator**
  - Validate and test architecture functionality
  - Modelling multi-FPGAs communication delays
  - SystemC allows higher or lower levels of abstraction (TLM, RTL-like)

- **Catapult High Level Synthesis (HLS)**
  - Direct transition from high-level simulator to RTL
  - Benefits: high level language, software emulation mode, early estimates on latency/throughput and FPGA-resources consumption → relatively easy validation, reprogrammability, and debug

- **Catapult HLS, over Vitis HLS**
  - Supports SystemC Synthesis
  - Vitis HLS C++ Software Emulation is not intuitevily applicable to a multi-FPGA simulation model
  - SystemC RTL-like flexibility is useful to model architectural details such as synchronization and feedback between hardware blocks

Architecture Design

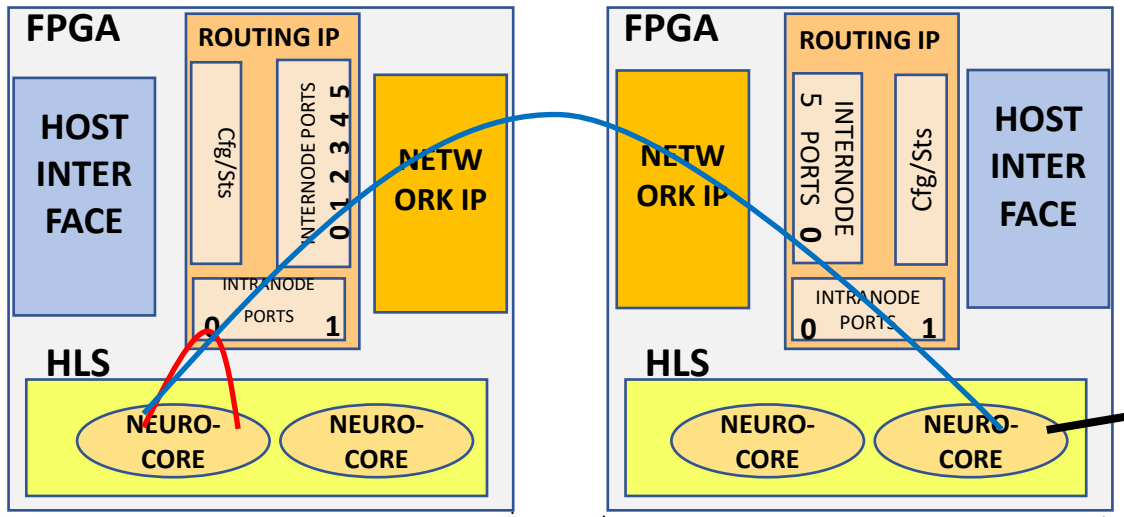
INFN Comm. IP

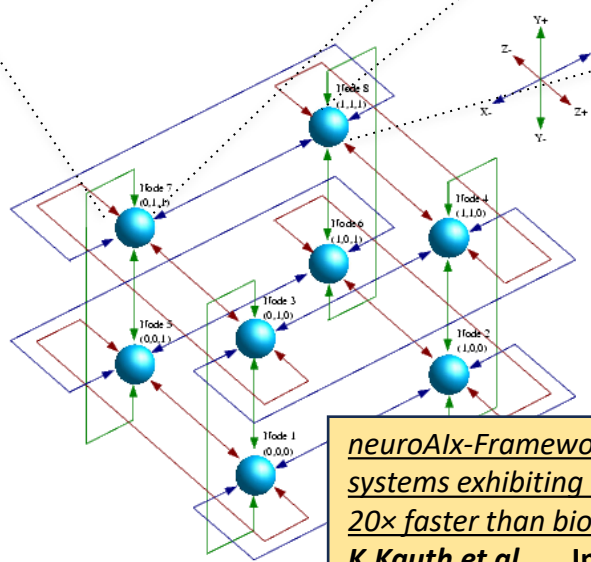


- Direct network of interconnected FPGAs
- Baseline topology assumption: 3D Torus
- Relying on internal INFN Communication IP for low-latency inter-node and intra-node communication (< 1 us for up to 1 kB packets)
  - Node = FPGA
  - Inter-node and intra-node communication between computing tasks (NEURO-CORE)
- NEURO-CORE
  - Core architectural component, containing a set of neurons
  - Exchanges spikes and barrier synchronization messages with other Neuro-Cores through the INFN Comm.IP
  - Developed with High Level Synthesis (HLS) tools
  - Switching to lower level VHDL if needed for performance of critical blocks

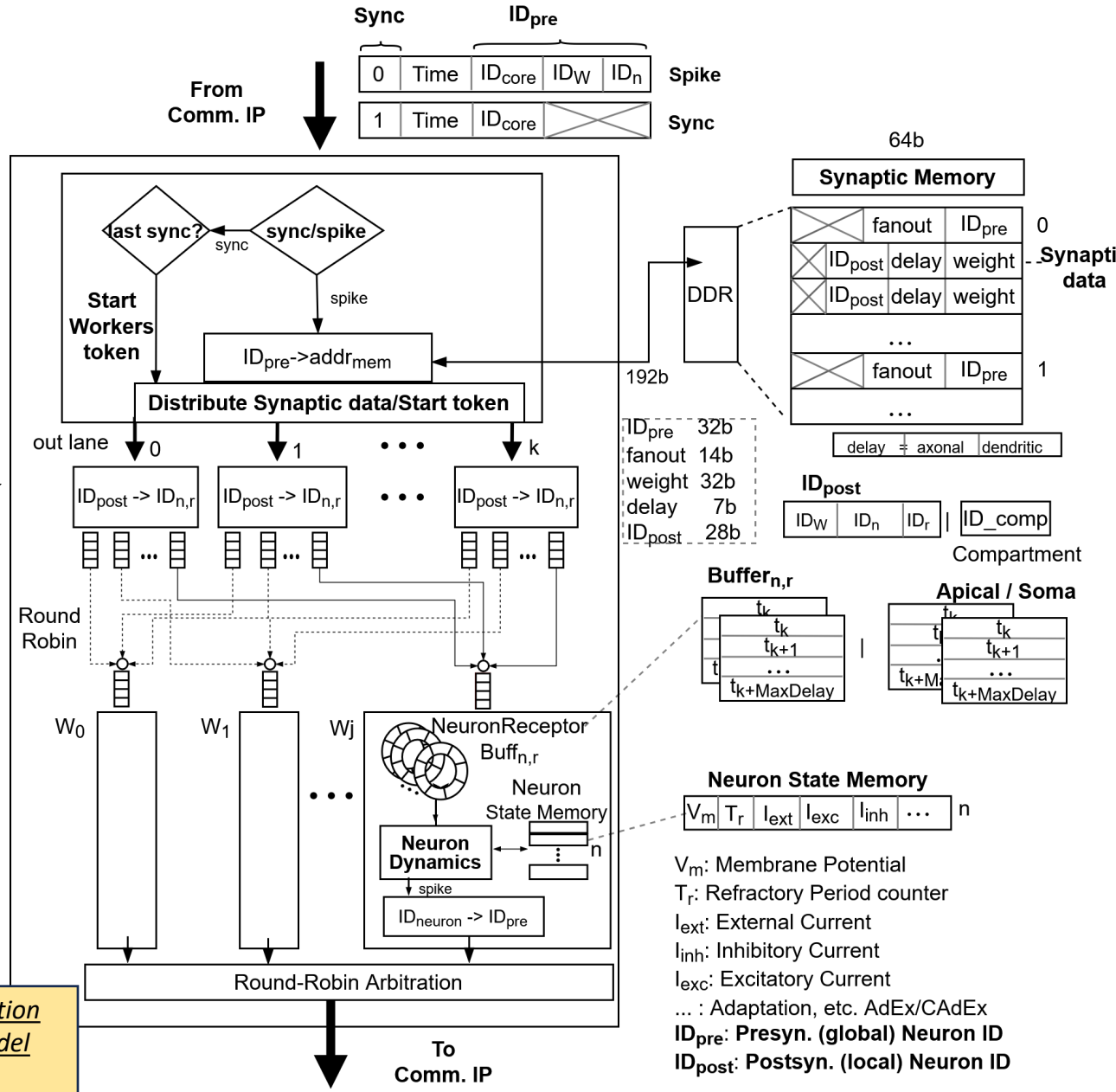
Architecture Design

INFN Comm. IP





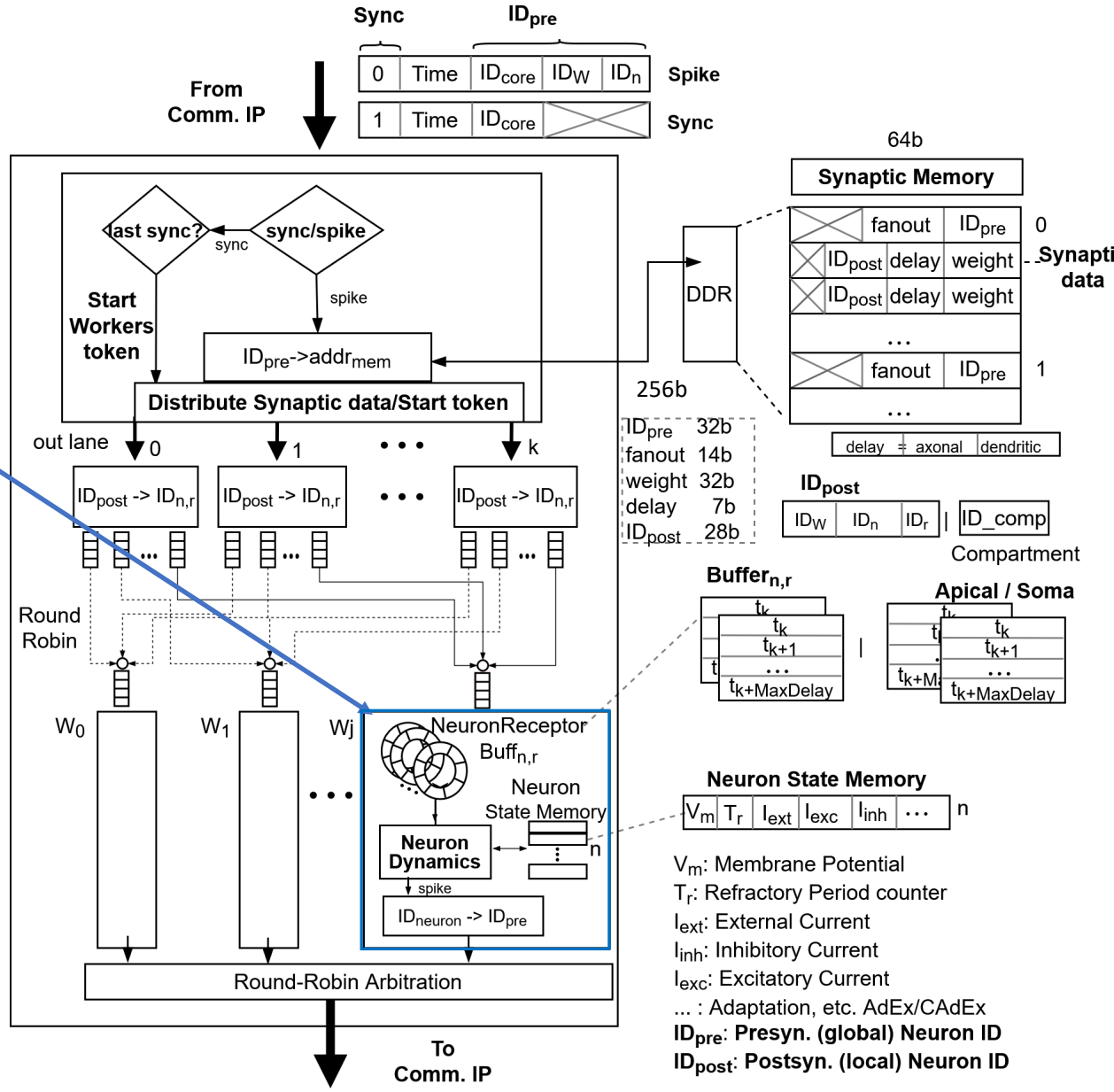
neuroAIx-Framework: design of future neuroscience simulation systems exhibiting execution of the cortical microcircuit model 20× faster than biological real-time (2023)
K.Kauth et al. **Inspiration for our first design**



Architecture Design – Neuro-Core

Modular architecture

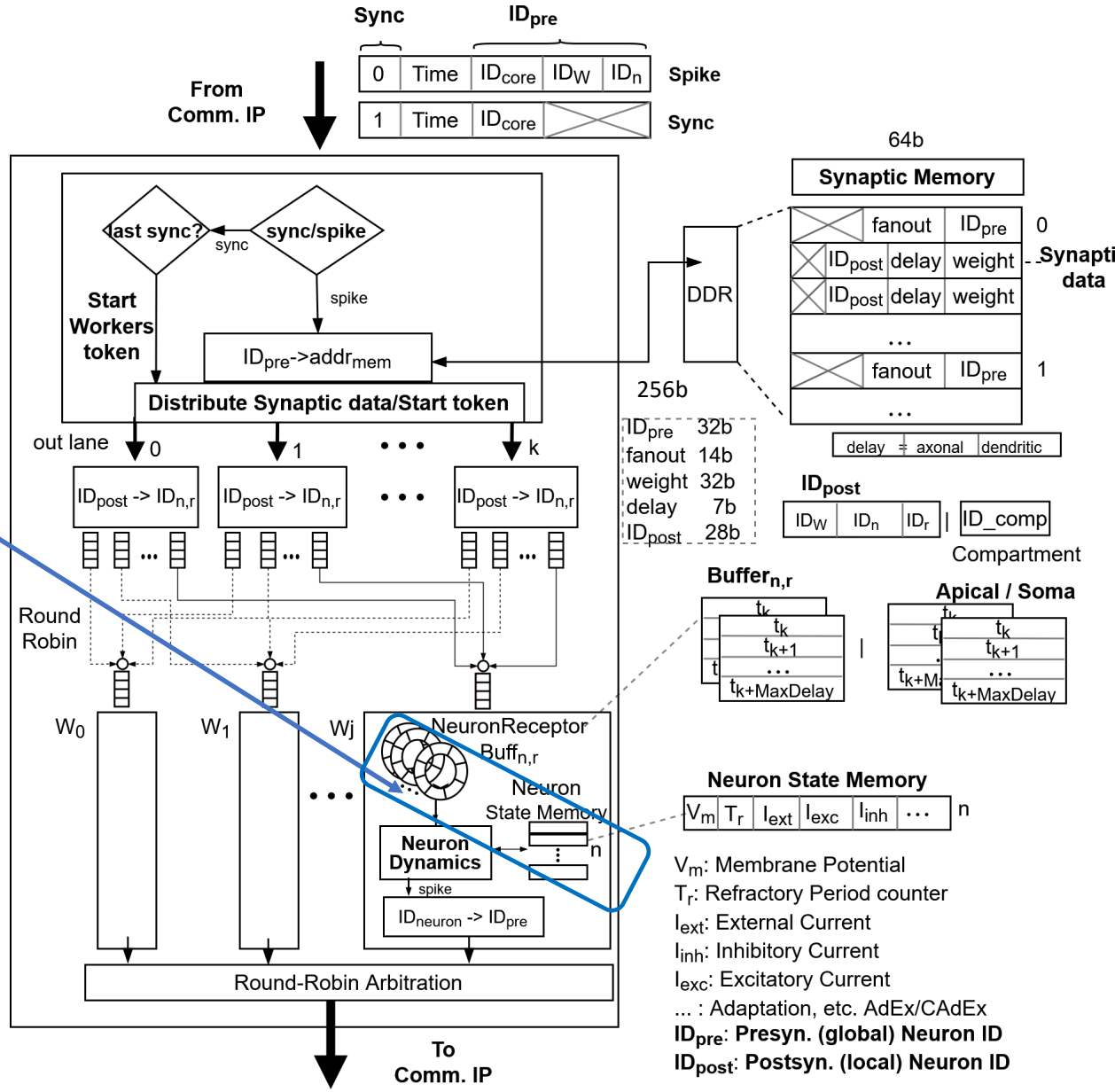
- **Neuro-Core contains Workers (Wj)**
- Workers contain neurons



Architecture Design – Neuro-Core

Modular architecture

- Neuro-Core contains Workers (Wj)
- **Workers contain neurons (n)**
  – each with multiple associated neuro-receptors (r) to model biologically realistic dynamics and multi-compartments



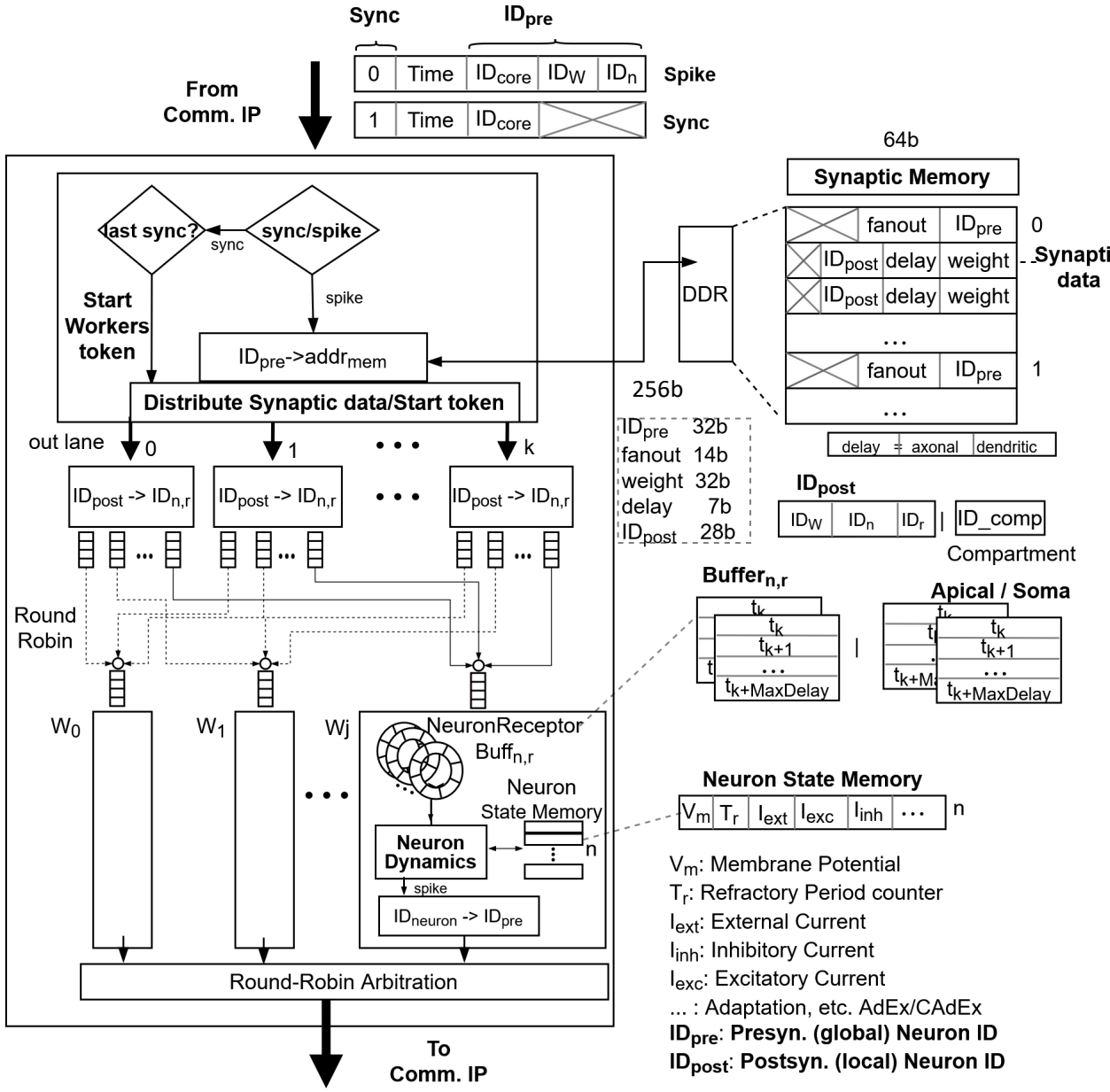
Architecture Design – Neuro-Core

Modular architecture

- Neuro-Core contains Workers (Wj)
- Workers contain neurons (n)

Reconfigurable

- **Optimal Bit-filed sizes and parameters (Nr of Workers, Nr of Neurons per Worker) will depend on target application**
- Synaptic Memory can be implemented as off-chip DRAM or local BRAM, according to size of target network connectome



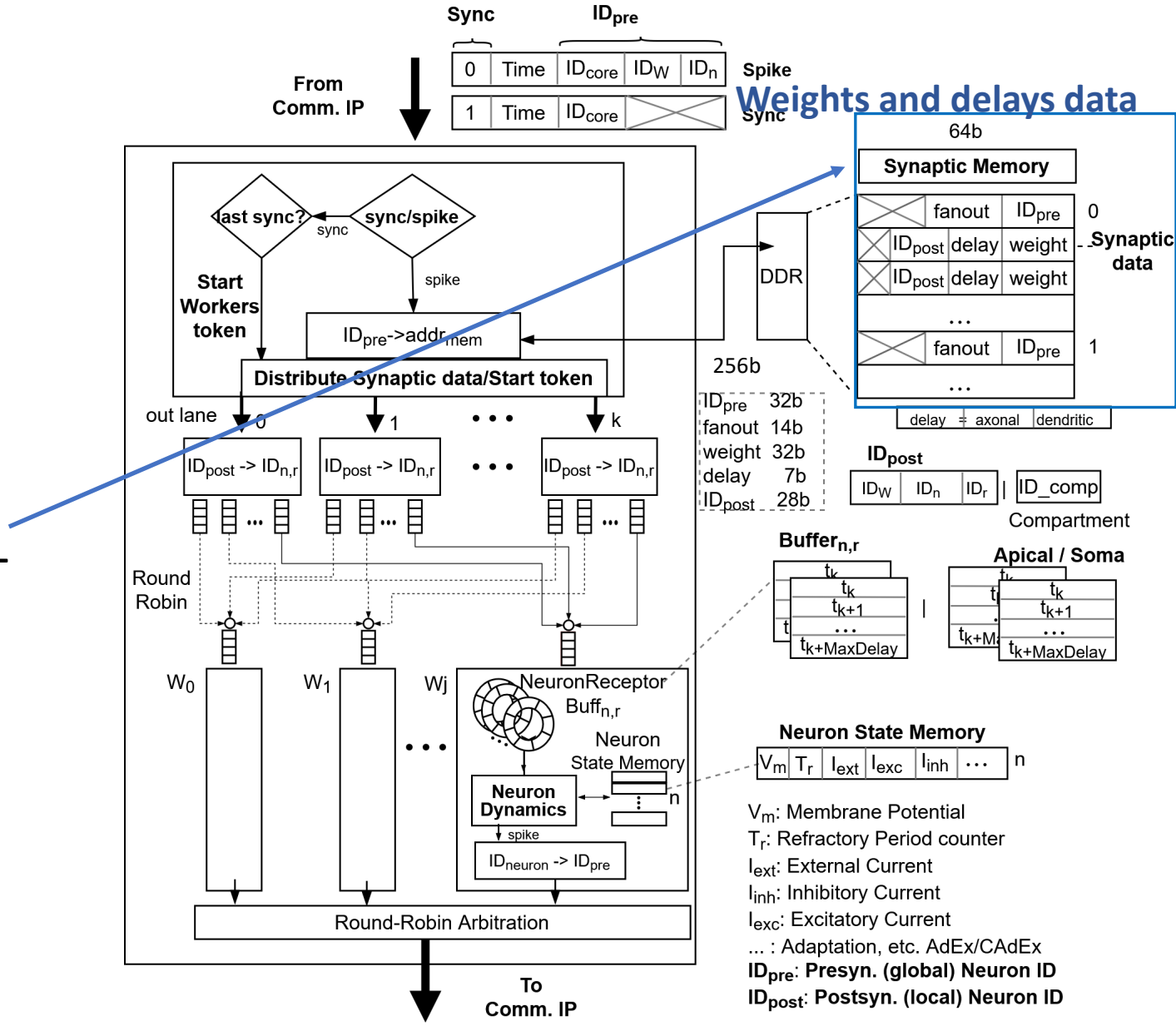
Architecture Design – Neuro-Core

Modular architecture

- Neuro-Core contains Workers (Wj)
- Workers contain neurons (n)

Reconfigurable

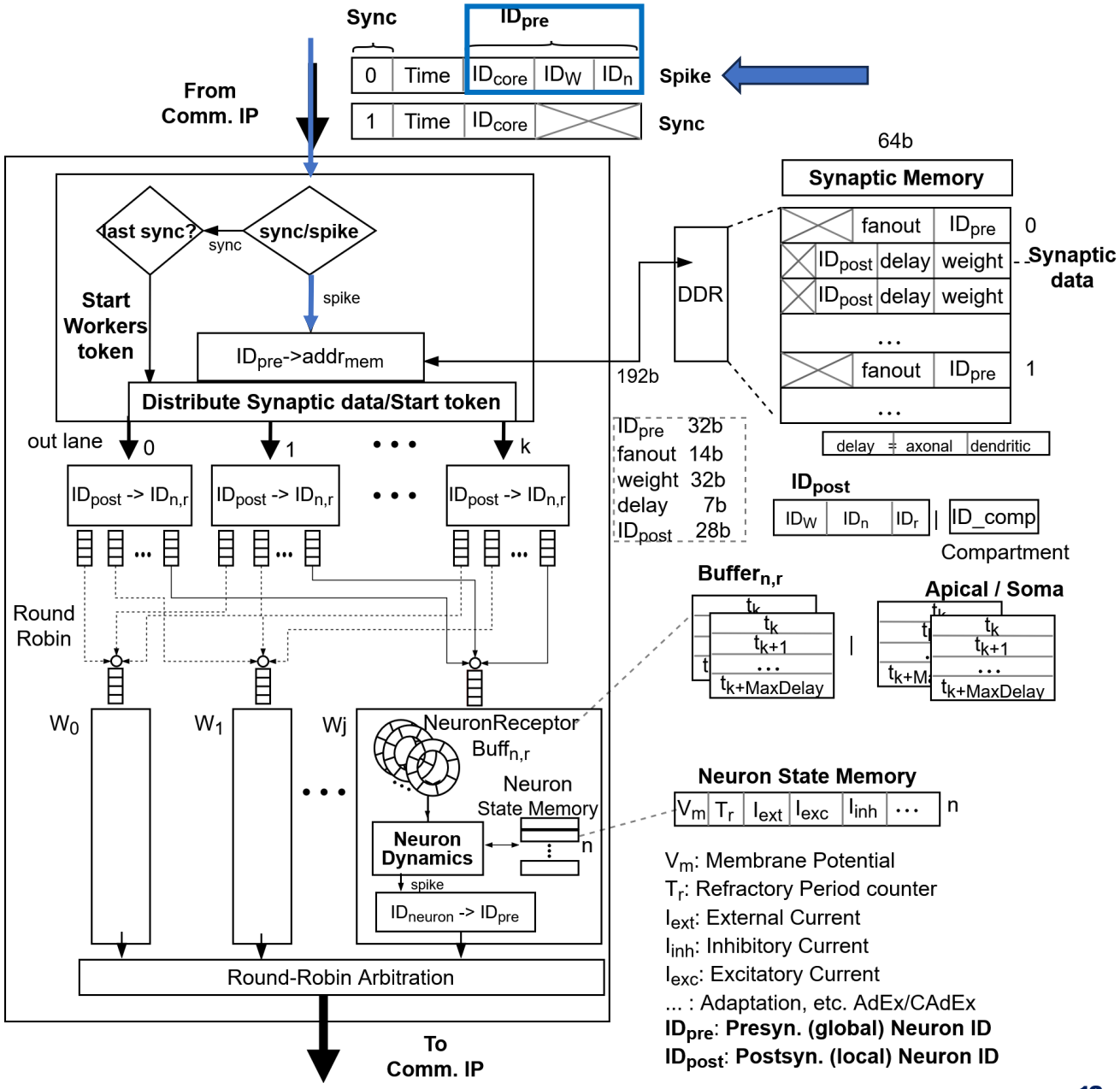
- Optimal Bit-filed sizes and parameters (Nr of Workers, Nr of Neurons per Worker) will depend on target application
- **Synaptic Memory can be implemented as off-chip DRAM or local BRAM, according to size of target network connectome**



Dataflow

Spike input

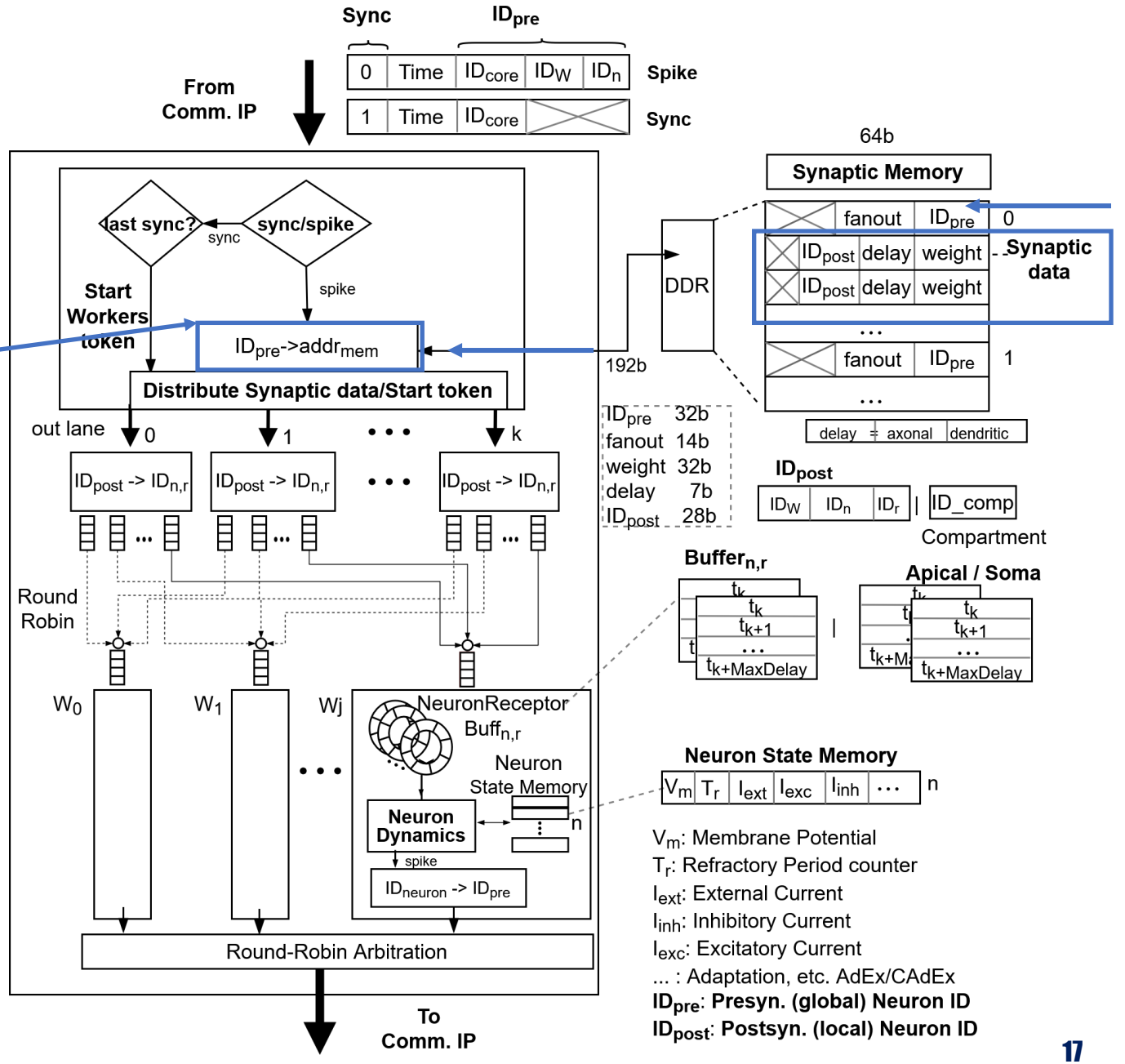
- Characterized by ID of pre-synaptic neuron who sent it
  Address Event Representation (AER)



Dataflow

Memory Look-Up and distribution

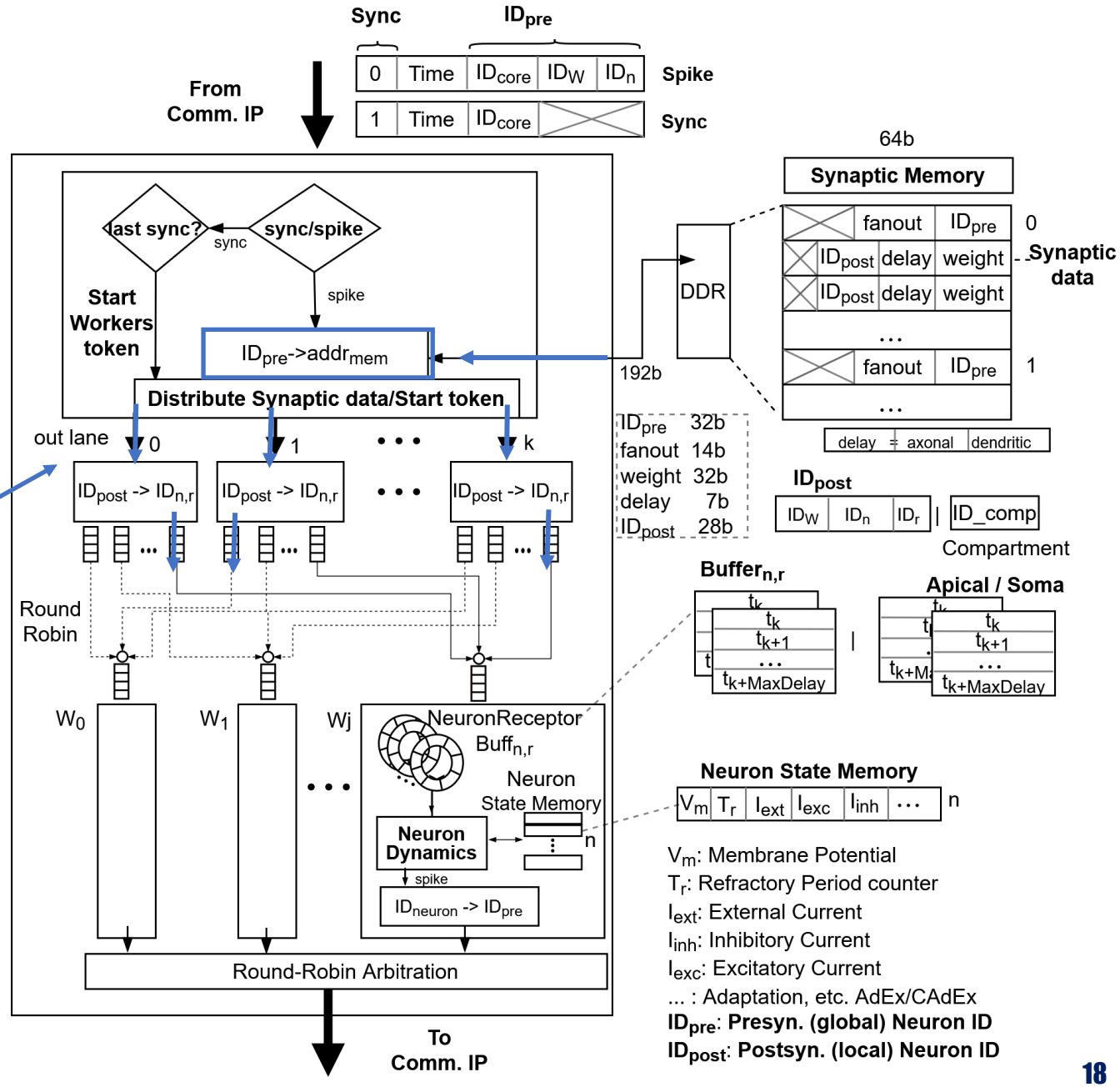
- Memory Look-Up according to **ID_{pre}**
  - **Extract the weight and delay of synapses connecting the ID_{pre} neuron to the post-synaptic neurons on this core**



Dataflow

Memory Look-Up and distribution

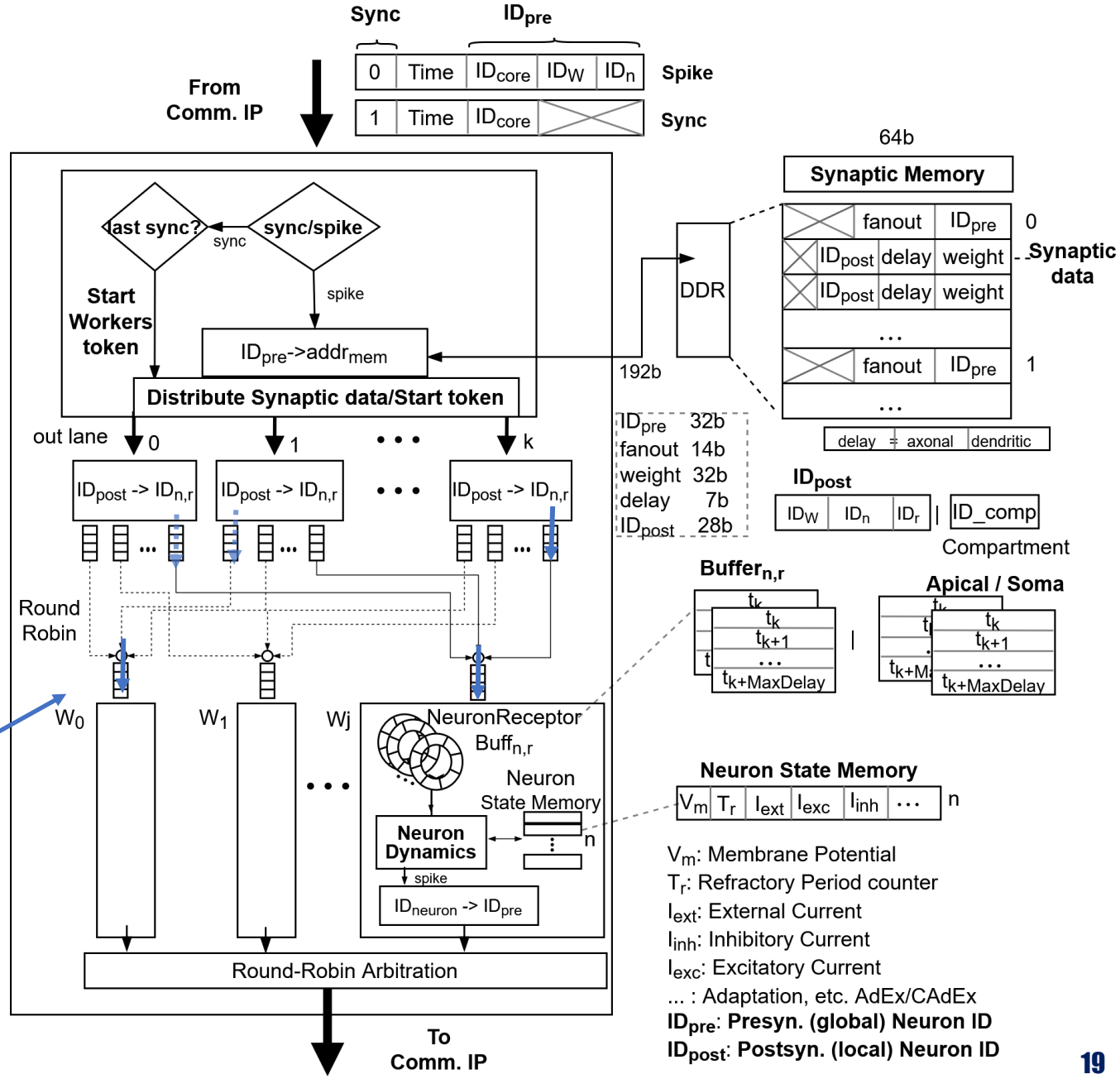
- Memory Look-Up according to **ID_{pre}**
  - Extract the weight and delay of synapses connecting the ID_{pre} neuron to the post-synaptic neurons on this core
- **Distribute this data to the post-synaptic neurons**
  - Multiple out lanes to increase throughput



Dataflow

Memory Look-Up and distribution

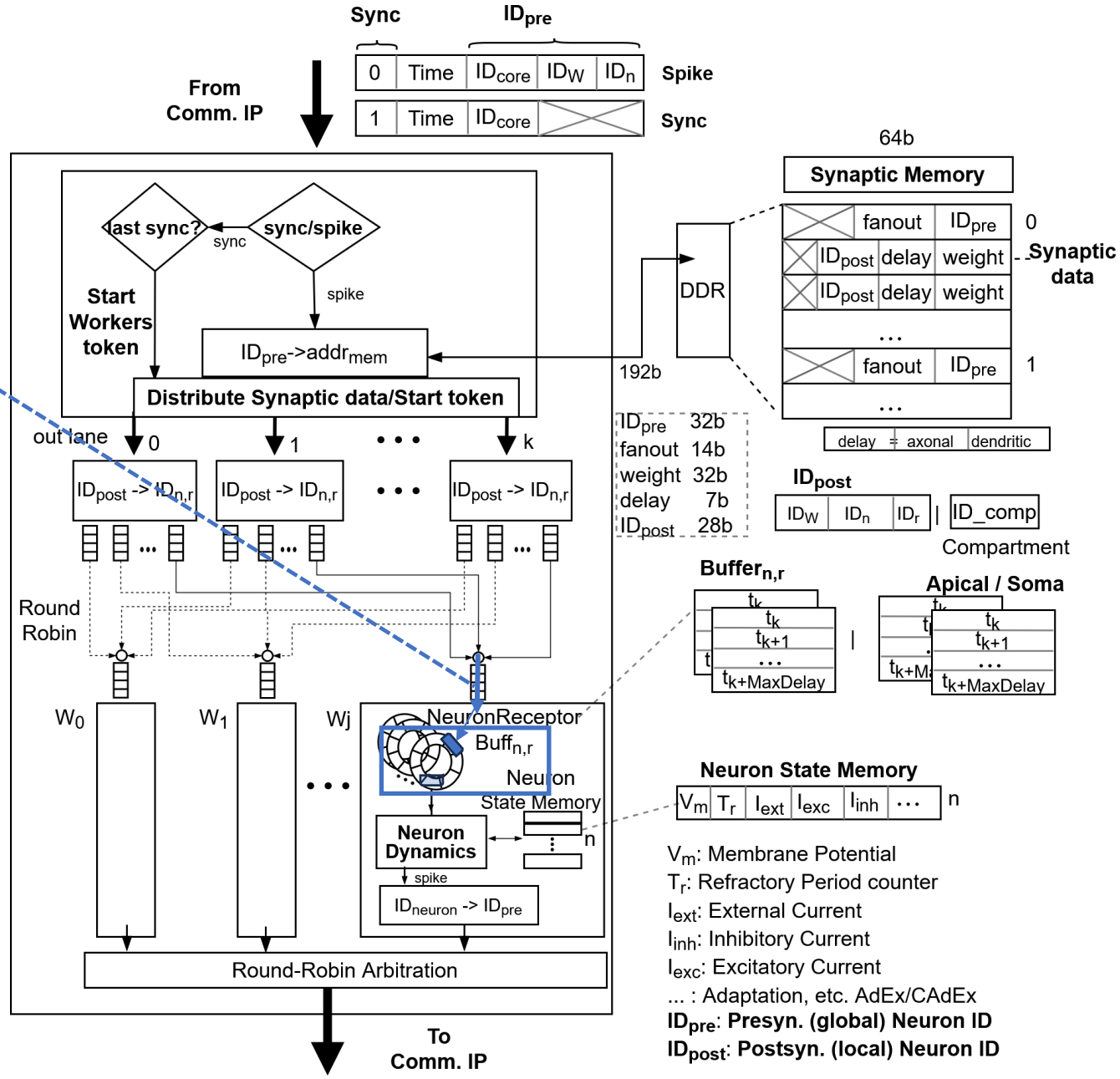
- Memory Look-Up according to **ID_{pre}**
  - Extract the weight and delay of synapses connecting the ID_{pre} neuron to the post-synaptic neurons on this core
- Distribute this data to the post-synaptic neurons
  - Multiple out lanes to increase throughput
  - **Load balancing Round Robin distribution to local Workers containing post-synaptic neurons**



Dataflow

- When **spike synaptic data (weight,delay)** reaches here, the **weight gets collocated in ring buffer** according to **delay**, to be processed at a later time, mimicking synaptic delays

- **Summed to previous spikes' weights** already in that time slot, making for an aggregate weight



Dataflow

- When **all pre-synaptic spikes have been received**

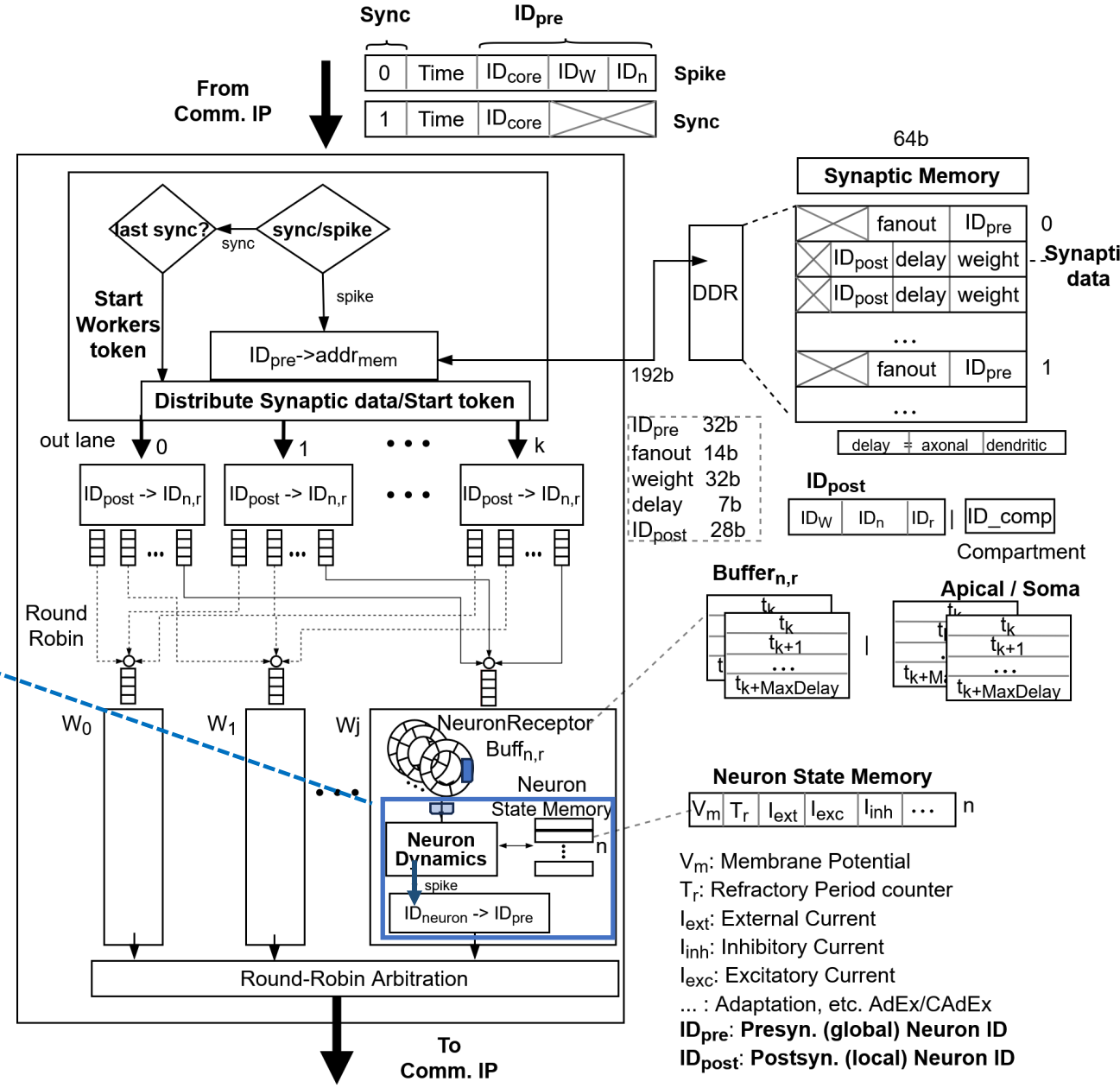
Worker can start Neurons Dynamics

Currently Leaky Integrate and Fire with delta shaped post synaptic currents (*iaf_psc_delta* from NEST simulator)

$$\frac{dV_\mathrm{m}}{dt} = -\frac{V_\mathrm{m} - E_\mathrm{L}}{\tau_\mathrm{m}} + \dot{\Delta}_\mathrm{syn} + \frac{I_\mathrm{syn} + I_\mathrm{e}}{C_\mathrm{m}}$$

$$\dot{\Delta}_\mathrm{syn}(t) = \sum_j w_j \sum_k \delta(t - t_j^k - d_j)$$

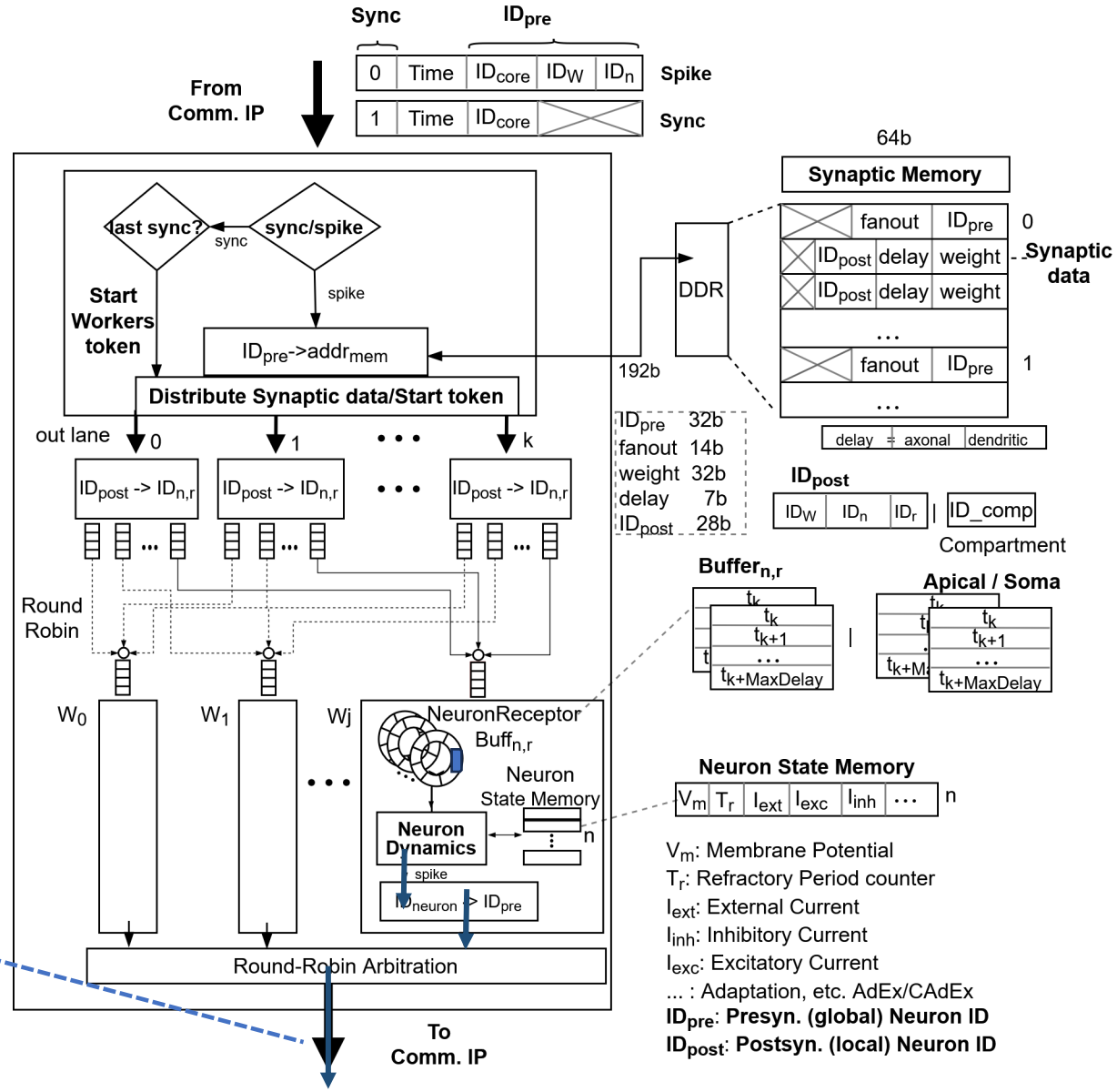
HLS allows agile study of different neuron models



Dataflow

Spike emission

- Broadcast to all Neuro-Cores

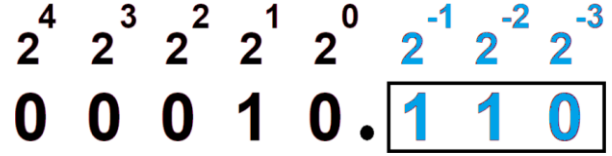


Functional Validation

- Comparing simulation results of our system with the enstablished NEST simulator on a common network testcase: same neuron model, connectivity, external input spike trains.

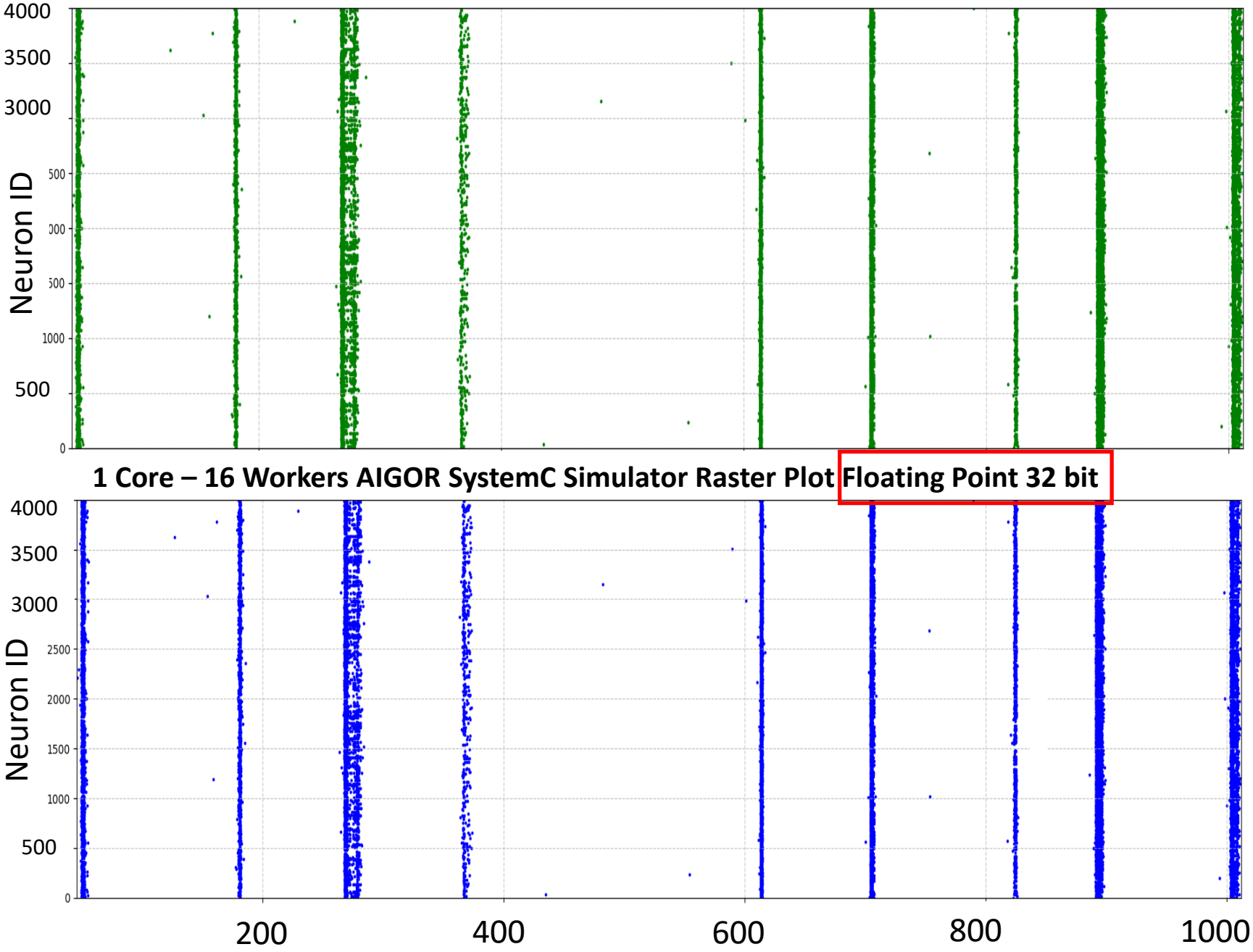
C++ simulator, with a Python interface, implementing a large number of models of biological neurons and synapses

- Bit-exact agreement expected for software-simulator but not for final hardware implementation for multiple reasons, e.g.:
  - 64-bit floating point arithmetic in NEST vs **fixed-point** numeric representation in our system; currently 32-bit, but evaluating smaller precision.



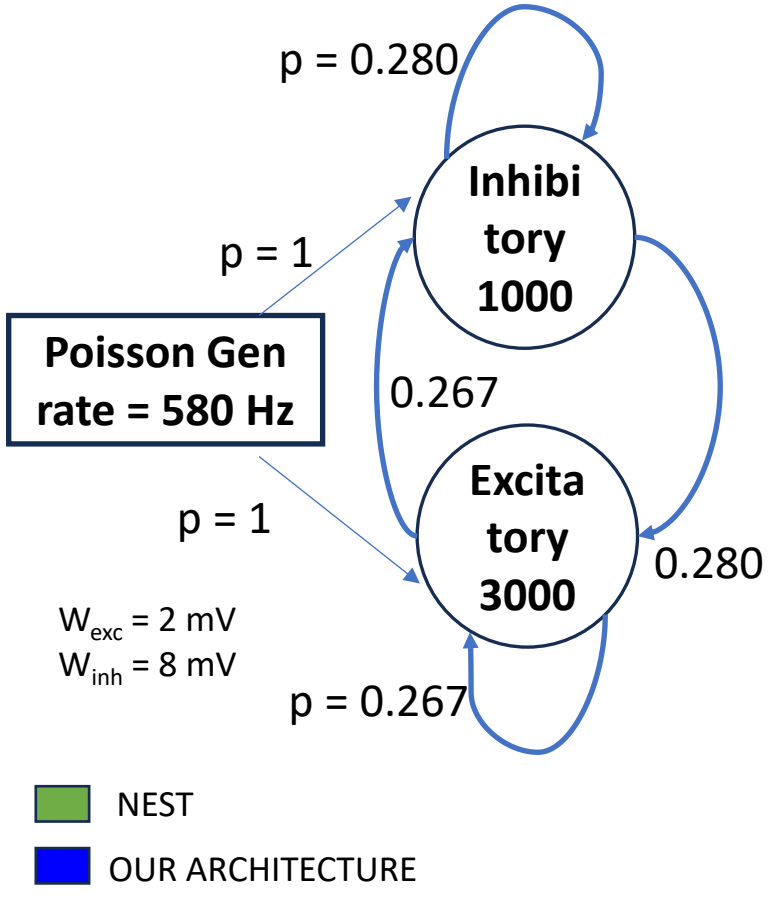
  - Neuron dynamics implementation: differential equations solver methods suited for software vs hardware-optimized methods
  - MPI communication vs custom communication

Functional Validation

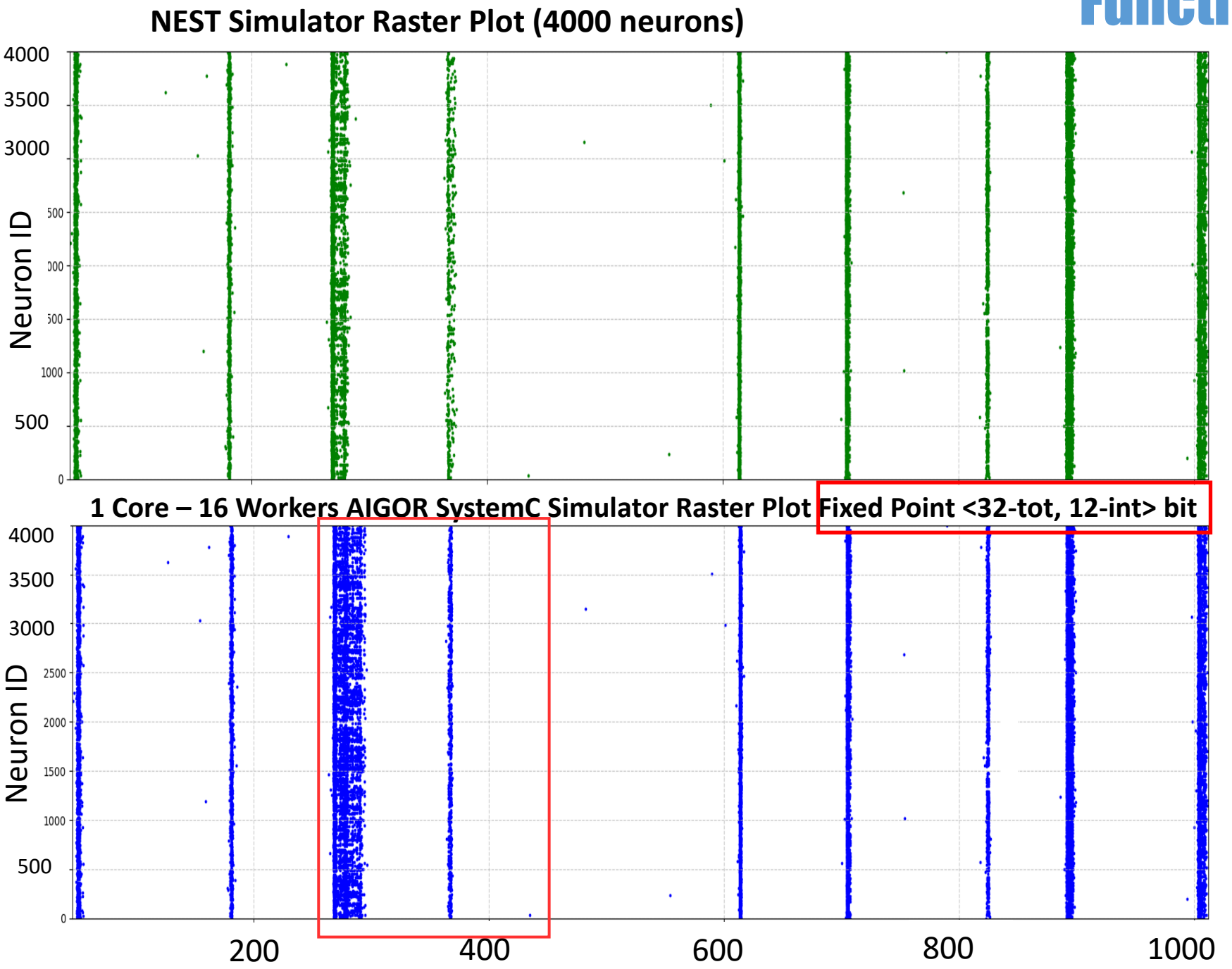


Test network

Brunel-style balanced random network with Poisson external spike input



Functional Validation



Studying effects of reduced fixed-point representation, using the SystemC RTL-like bit-accurate simulator

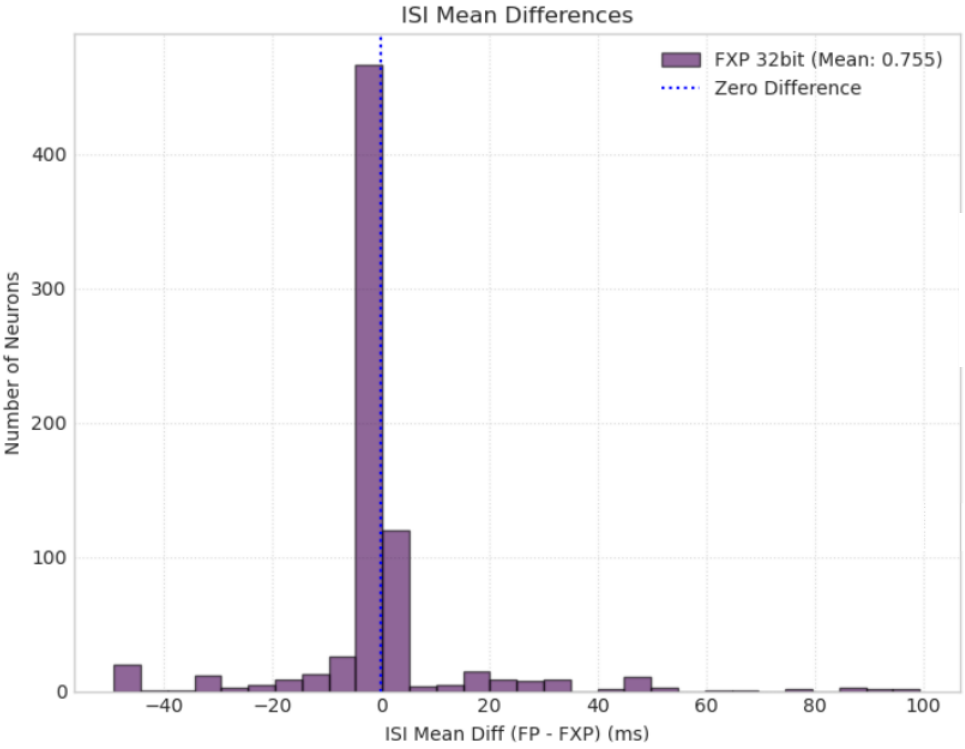
NEST

OUR ARCHITECTURE

Spike-train distance metrics

Studying effects of reduced fixed-point representation, using the SystemC RTL-like bit-accurate simulator

Neuron mean
Inter-spike Interval (ISI)
difference FP – FXP32

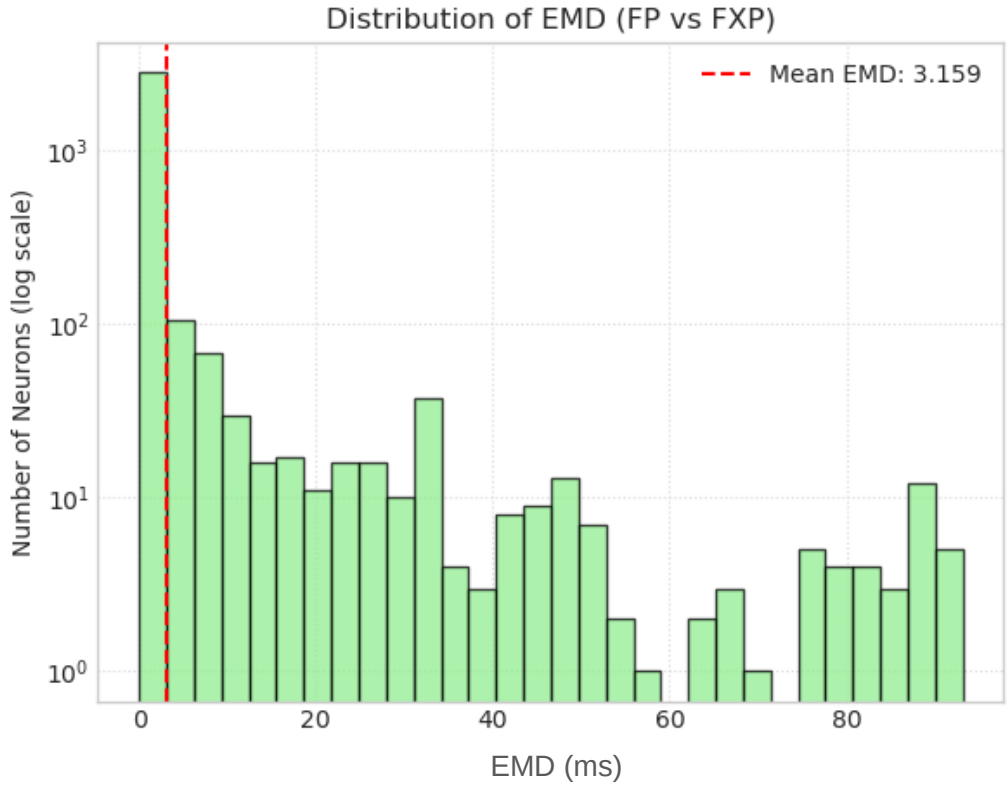


Earth Mover's Distance (or Wasserstein Distance)

$$l_1(u, v) = \int_{-\infty}^{+\infty} |U - V|$$

u and v : spike-time prob distribution

U and V : CDFs of u and v



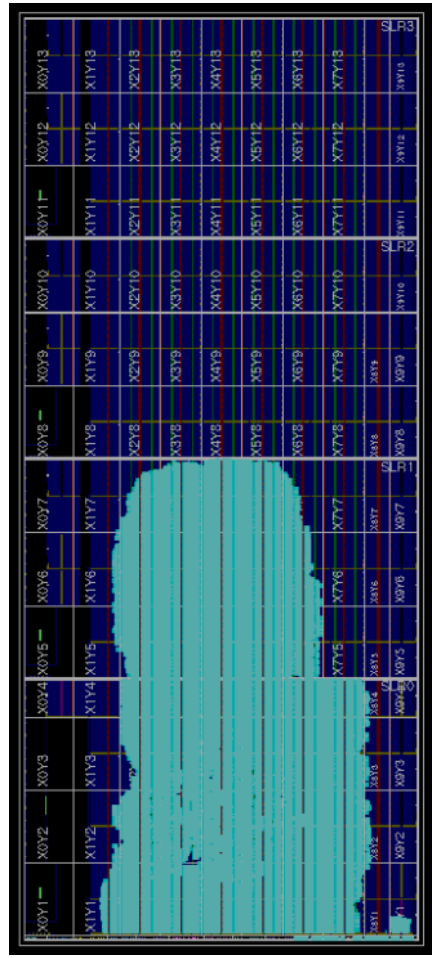
FPGA Occupancy Single Neuro-core on a Versal VPK-180

W = workers

n = neurons

VPK-180 Total PL mem 994 Mb

Single Neuro-Core	LUT	FF	DSP	BRAM	Synaptic Mem requires
16 n (4 W, 4 n x W)	13.5k **0.40%**	21.6k **0.32%**	0 **0.00%**	2 **0.04%**	16 Kb
64 n (4 W, 16 n x W)	21.1k **0.63%**	31.4k **0.47%**	0 **0.00%**	4 **0.08%**	197 Kb
512 n (4 W, 128 n x W)	75.5k **2.25%**	30.6k **0.45%**	0 **0.00%**	16 **0.32%**	4 Mb
2048 n (32 W, 64 n x W)	764.0k **22.7%**	197.3k **2.93%**	2 **0.01%**	96 **1.94%**	84 Mb
2048 n (16 W, 128 n x W)	708.5k **21.1%**	106.9k **1.59%**	34 **0.24%**	64 **1.30%**	84 Mb



- Currently validating complete hardware testbed, with a Neuro-Core + a Poisson input Core on a single FPGA

Next steps

- Begin deployment of a target application, starting with a single FPGA
- Optimize fixed-point representation, architecture parameters and neuron model alongside the target application performance
- Validate multi-FPGA design